Московский физико-технический институт
(государственный университет)

На правах рукописи
УДК 519.254

Исаченко Роман Владимирович

Выбор модели декодирования сигналов в пространствах высокой размерности

05.13.17 — Теоретические основы информатики

Диссертация на соискание ученой степени
кандидата физико-математических наук

Научный руководитель:
д.ф.-м.н. В. В. Стрижов

Москва — 2019

Оглавление

Стр.

Введение

Диссертационная работа посвящена построению математических моделей машинного обучения в пространствах высокой размерности. Разработанные методы позволяют учесть зависимости, имеющиеся в исходных данных, с целью построения простой и устройчивой модели.

Актуальность темы.

В работе исследуется задача декодирования сигналов. При построении машинного обучения возникает необходимость построения низкоразмерного признакого пространства. Требуется по входному исходному сигналу предсказать отклик на этот сигнал.

Сложностью задачи является избыточность исходного описания данных. Исходное признаковое пространство является мультикоррелированным. Финальная предсказательная модель оказывается неустойчивой. Для построения простой, устойчивой модели применяются методы снижения размерности пространства [1, 2] и выбора признаков [3, 4].

В работе рассматриваются задачи с векторной целевой переменной. При предсказании векторной целевой переменной возникает необходимости в анализе структуры целевого пространства. Целевое пространство содержит зависимости. В работе предлагаются методы, которые позволяют учесть зависимости как в исходном пространстве объектов, так и в пространстве целевой переменной.

параграф про снижение размерности

параграф про выбор признаков

параграф про метрическое обучение

Цели работы.

Задачи работы.

Основные положения, выносимые на защиту.

1. Метод снижения размерности пространства, отображающий независимую и целевую переменные в единое скрытое низкоразмерное представление.
2. Методы выбора признаков для задач с многомерной целевой переменной, учитывающие структуры пространств.
3. Алгоритм выбора наиболее влиятельных параметров для оптимизации нелинейной модели.
4. Алгоритм метрического обучения для временных рядов с процедурой их выравнивания.
5. Программный комплекс, включающий прогностические модели для высокоразмерных данных. Проведены вычислительные эксперименты, подтверждающие адекватность методов.

Методы исследования.

Научная новизна.

Теоретическая значимость.

Практическая значимость.

Степень достоверности и апробация работы.

Публикации по теме диссертации.

Структура и объем работы.

Личный вклад. Все приведенные результаты, кроме отдельно оговоренных случаев, по- лучены диссертантом лично при научном руководстве д.ф.-м.н. В. В. Стрижова.

Краткое содержание работы по главам.

Глава 1
Постановка задачи декодирования

1.1. Основные определения

1.2. Задача восстановления регрессии

Цель задачи регрессии – построить прогноз целевой переменной $\mathbf{y} \in \mathbb{R}^r$ с r компонентами по набору независимых переменных $\mathbf{x} \in \mathbb{R}^n$, где n – число признаков. Предполагается, что существует линейная зависимость между объектом $\mathbf{x}$ и целевой переменной $\mathbf{y}$ следующего вида:

$$\mathbf{y} = \boldsymbol{\Theta}\mathbf{x} + \boldsymbol{\varepsilon}. \tag{1.1}$$

Здесь $\boldsymbol{\Theta} \in \mathbb{R}^{r \times n}$ – матрица параметров модели, $\boldsymbol{\varepsilon} \in \mathbb{R}^r$ представляет собой вектор остатков. Необходимо найти матрицу параметров модели $\boldsymbol{\Theta}$ при известном наборе данных $(\mathbf{X}, \mathbf{Y})$, где $\mathbf{X} \in \mathbb{R}^{m \times n}$ – матрица плана, $\mathbf{Y} \in \mathbb{R}^{m \times r}$ – целевая матрица:

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^\intercal = [\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_n]; \quad \mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_m]^\intercal = [\boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_r].$$

Столбцы $\boldsymbol{\chi}_j$ матрицы $\mathbf{X}$ являются признаками объекта, столбцы $\boldsymbol{\nu}_j$ матрицы $\mathbf{Y}$ являются целевыми столбцами.

Оптимальные параметры определяются минимизацией функции ошибки. Определим квадратичную функцию потерь следующим образом:

$$\mathcal{L}(\boldsymbol{\Theta}, \mathbf{X}, \mathbf{Y}) = \left\| \underset{m \times r}{\mathbf{Y}} - \underset{m \times n}{\mathbf{X}} \cdot \underset{r \times n}{\boldsymbol{\Theta}}^\intercal \right\|_2^2 \to \min_{\boldsymbol{\Theta}}. \tag{1.2}$$

Решением (1.2) является следующая матрица:

$$\boldsymbol{\Theta} = \mathbf{Y}^\intercal \mathbf{X} (\mathbf{X}^\intercal \mathbf{X})^{-1}.$$

Наличие линейной зависимости между столбцами матрицы $\mathbf{X}$ приводит к неустойчивому решению задачи оптимизации (1.2). Если существует вектор $\boldsymbol{\alpha} \neq \mathbf{0}_n$ такой, что $\mathbf{X}\boldsymbol{\alpha} = \mathbf{0}_m$, то добавление $\boldsymbol{\alpha}$ в любой столбец матрицы $\boldsymbol{\Theta}$

не меняет значение функции потерь $\mathcal{L}(\boldsymbol{\Theta}, \mathbf{X}, \mathbf{Y})$. В этом случае матрица $\mathbf{X}^\intercal\mathbf{X}$ близка к сингулярной и не обратима. Чтобы избежать сильной линейной зависимости, используются методы снижения размерности и выбора признаков.

1.3. Метод проекции в скрытое пространство

Для устранения линейной зависимости и снижения размерности входного пространства объектов широко используется метод главных компонент (PCA). Основным недостатком метода PCA является отсутствие взаимосвязи между признаками и целевыми векторами. Алгоритм частичных наименьших квадратов проецирует матрицу плана $\mathbf{X}$ и целевую матрицу $\mathbf{Y}$ в скрытое пространство с малой размерностью ($l < n$). Алгоритм PLS находит в скрытом пространстве матрицы $\mathbf{T}, \mathbf{U} \in \mathbb{R}^{m \times l}$, которые лучше всего описывают оригинальные матрицы $\mathbf{X}$ и $\mathbf{Y}$, и учитывает взаимосвязь между ними.

Матрица плана $\mathbf{X}$ и целевая матрица $\mathbf{Y}$ проецируются в скрытое пространство следующим образом:

$$\underset{m\times n}{\mathbf{X}} = \underset{m\times l}{\mathbf{T}} \cdot \underset{l\times n}{\mathbf{P}^\intercal} + \underset{m\times n}{\mathbf{F}} = \sum_{k=1}^{l} \underset{m\times 1}{\mathbf{t}_k} \cdot \underset{1\times n}{\mathbf{p}_k^\intercal} + \underset{m\times n}{\mathbf{F}}, \tag{1.3}$$

$$\underset{m\times r}{\mathbf{Y}} = \underset{m\times l}{\mathbf{U}} \cdot \underset{l\times r}{\mathbf{Q}^\intercal} + \underset{m\times r}{\mathbf{E}} = \sum_{k=1}^{l} \underset{m\times 1}{\mathbf{u}_k} \cdot \underset{1\times r}{\mathbf{q}_k^\intercal} + \underset{m\times r}{\mathbf{E}}, \tag{1.4}$$

где $\mathbf{T}$ и $\mathbf{U}$ – образы исходных матриц в скрытом пространстве, причём столбцы матрицы $\mathbf{T}$ ортогональны; $\mathbf{P}$ и $\mathbf{Q}$ – матрицы перехода; $\mathbf{E}$ и $\mathbf{F}$ – матрицы остатков. Алгоритм PLS максимизирует линейную зависимость между столбцами матриц $\mathbf{T}$ и $\mathbf{U}$

$$\mathbf{U} \approx \mathbf{TB}, \quad \mathbf{B} = \operatorname{diag}(\beta_k), \quad \beta_k = \mathbf{u}_k^\intercal \mathbf{t}_k / (\mathbf{t}_k^\intercal \mathbf{t}_k).$$

Псевдокод метода регрессии PLS приведен в алгоритме 1. Алгоритм итеративно на каждом из l шагов вычисляет по одному столбцу $\mathbf{t}_k$, $\mathbf{u}_k$, $\mathbf{p}_k$, $\mathbf{q}_k$ матриц $\mathbf{T}$, $\mathbf{U}$, $\mathbf{P}$, $\mathbf{Q}$ соответственно. После вычисления следующего набора векторов

из матриц $\mathbf{X}$, $\mathbf{Y}$ вычитаются очередные одноранговые аппроксимации. Первым шагом необходимо произвести нормировку столбцов исходных матриц (вычесть среднее и разделить на стандартное отклонение). На этапе тестирования необходимо провести нормировку тестовых данных, вычислить предсказание модели 1.1, а затем провести обратную нормировку.

Algorithm 1 Алгоритм PLSR

Вход: $\mathbf{X}, \mathbf{Y}, l$;

Выход: $\mathbf{T}, \mathbf{P}, \mathbf{Q}$;

1: нормировать матрицы $\mathbf{X}$ и $\mathbf{Y}$ по столбцам
2: инициализировать $\mathbf{u}_0$ (первый столбец матрицы $\mathbf{Y}$)
3: $\mathbf{X}_1 = \mathbf{X}; \mathbf{Y}_1 = \mathbf{Y}$
4: **для** $k = 1, \ldots, l$
5: **повторять**
6: $\mathbf{w}_k := \mathbf{X}_k^\intercal \mathbf{u}_{k-1} / (\mathbf{u}_{k-1}^\intercal \mathbf{u}_{k-1}); \quad \mathbf{w}_k := \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$
7: $\mathbf{t}_k := \mathbf{X}_k \mathbf{w}_k$
8: $\mathbf{c}_k := \mathbf{Y}_k^\intercal \mathbf{t}_k / (\mathbf{t}_k^\intercal \mathbf{t}_k); \quad \mathbf{c}_k := \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|}$
9: $\mathbf{u}_k := \mathbf{Y}_k \mathbf{c}_k$
10: **пока** $\mathbf{t}_k$ не стабилизируется
11: $\mathbf{p}_k := \mathbf{X}_k^\intercal \mathbf{t}_k / (\mathbf{t}_k^\intercal \mathbf{t}_k)$, $\mathbf{q}_k := \mathbf{Y}_k^\intercal \mathbf{t}_k / (\mathbf{t}_k^\intercal \mathbf{t}_k)$
12: $\mathbf{X}_{k+1} := \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^\intercal$
13: $\mathbf{Y}_{k+1} := \mathbf{Y}_k - \mathbf{t}_k \mathbf{q}_k^\intercal$

Вектора $\mathbf{t}_k$ и $\mathbf{u}_k$ из внутреннего цикла алгоритма 1 содержат информацию о матрице объектов $\mathbf{X}$ и матрице ответов $\mathbf{Y}$ соответственно. Блоки из шагов (6)-(7) и шагов (8)-(9) — аналоги алгоритма PCA для матриц $\mathbf{X}$ и $\mathbf{Y}$ [5]. Последовательное выполнение блоков позволяет учесть взаимную связь между матрицами $\mathbf{X}$ и $\mathbf{Y}$.

Теоретическое обоснование алгоритма PLS следует из следующих утвержде-

ний.

Утверждение 1. Наилучшее описание матриц $\mathbf{X}$ и $\mathbf{Y}$ с учётом их взаимосвязи достигается при максимизации ковариации между векторами $\mathbf{t}_k$ и $\mathbf{u}_k$.

Доказательство. Утверждение следует из равенства

$$\operatorname{cov}(\mathbf{t}_k, \mathbf{u}_k) = \operatorname{corr}(\mathbf{t}_k, \mathbf{u}_k) \cdot \sqrt{\operatorname{var}(\mathbf{t}_k)} \cdot \sqrt{\operatorname{var}(\mathbf{u}_k)}.$$

Максимизация дисперсий векторов $\mathbf{t}_k$ и $\mathbf{u}_k$ отвечает за сохранение информации об исходных матрицах, корреляция между векторами отвечает взаимосвязи между $\mathbf{X}$ и $\mathbf{Y}$. □

Во внутреннем цикле алгоритма вычисляются нормированные вектора весов $\mathbf{w}_k$ и $\mathbf{c}_k$. Из данных векторов строятся матрицы весов $\mathbf{W}$ и $\mathbf{C}$ соответственно.

Утверждение 2. В результате выполнения внутреннего цикла вектора $\mathbf{w}_k$ и $\mathbf{c}_k$ будут собственными векторами матриц $\mathbf{X}_k^\intercal \mathbf{Y}_k \mathbf{Y}_k^\intercal \mathbf{X}_k$ и $\mathbf{Y}_k^\intercal \mathbf{X}_k \mathbf{X}_k^\intercal \mathbf{Y}_k$, соответствующими максимальным собственным значениям.

$$\mathbf{w}_k \propto \mathbf{X}_k^\intercal \mathbf{u}_{k-1} \propto \mathbf{X}_k^\intercal \mathbf{Y}_k \mathbf{c}_{k-1} \propto \mathbf{X}_k^\intercal \mathbf{Y}_k \mathbf{Y}_k^\intercal \mathbf{t}_{k-1} \propto \mathbf{X}_k^\intercal \mathbf{Y}_k \mathbf{Y}_k^\intercal \mathbf{X}_k \mathbf{w}_{k-1},$$

$$\mathbf{c}_k \propto \mathbf{Y}_k^\intercal \mathbf{t}_k \propto \mathbf{Y}_k^\intercal \mathbf{X}_k \mathbf{w}_k \propto \mathbf{Y}_k^\intercal \mathbf{X}_k \mathbf{X}_k^\intercal \mathbf{u}_{k-1} \propto \mathbf{Y}_k^\intercal \mathbf{X}_k \mathbf{X}_k^\intercal \mathbf{Y}_k \mathbf{c}_{k-1},$$

где символ $\propto$ означает равенство с точностью до мультипликативной константы.

Доказательство. Утверждение следует из того факта, что правила обновления векторов $\mathbf{w}_k$, $\mathbf{c}_k$ совпадают с итерацией алгоритма поиска максимального собственного значения. Данный алгоритм основан на следующем факте.

Если матрица $\mathbf{A}$ диагонализуема, $\mathbf{x}$ — некоторый вектор, то

$$\lim_{k \to \infty} \mathbf{A}^k \mathbf{x} = \lambda_{\max}(\mathbf{A}) \cdot \mathbf{v}_{\max},$$

где $\lambda_{\max}(\mathbf{A})$ — максимальное собственное значение матрицы $\mathbf{A}$, $\mathbf{v}_{\max}$ — собственный вектор матрицы $\mathbf{A}$, соответствующий $\lambda_{\max}(\mathbf{A})$.

□

Утверждение 3. Обновление векторов по шагам (6)–(9) алгоритма 1 соответствует максимизации ковариации между векторами $\mathbf{t}_k$ и $\mathbf{u}_k$.

Доказательство. Максимальная ковариация между векторами $\mathbf{t}_k$ и $\mathbf{u}_k$ равна максимальному собственному значению матрицы $\mathbf{X}_k^\intercal \mathbf{Y}_k \mathbf{Y}_k^\intercal \mathbf{X}_k$:

$$\begin{aligned}\max_{\mathbf{t}_k,\mathbf{u}_k} \operatorname{cov}(\mathbf{t}_k,\mathbf{u}_k)^2 &= \max_{\substack{\|\mathbf{w}_k\|=1\\ \|\mathbf{c}_k\|=1}} \operatorname{cov}\left(\mathbf{X}_k\mathbf{w}_k, \mathbf{Y}_k\mathbf{c}_k\right)^2 = \max_{\substack{\|\mathbf{w}_k\|=1\\ \|\mathbf{c}_k\|=1}} \operatorname{cov}\left(\mathbf{c}_k^\intercal \mathbf{Y}_k^\intercal \mathbf{X}_k\mathbf{w}_k\right)^2 = \\ &= \max_{\|\mathbf{w}_k\|=1} \operatorname{cov}\left\|\mathbf{Y}_k^\intercal \mathbf{X}_k\mathbf{w}_k\right\|^2 = \max_{\|\mathbf{w}_k\|=1} \mathbf{w}_k^\intercal \mathbf{X}_k^\intercal \mathbf{Y}_k \mathbf{Y}_k^\intercal \mathbf{X}_k\mathbf{w}_k = \\ &= \lambda_{\max}\left(\mathbf{X}_k^\intercal \mathbf{Y}_k \mathbf{Y}_k^\intercal \mathbf{X}_k\right),\end{aligned}$$

где $\lambda_{\max}(\mathbf{A})$ — максимальное собственное значение матрицы $\mathbf{A}$. Применяя утверждение 2, получаем требуемое. □

После завершения внутреннего цикла на шаге (11) вычисляются вектора $\mathbf{p}_k$, $\mathbf{q}_k$ проецированием столбцов матриц $\mathbf{X}_k$ и $\mathbf{Y}_k$ на вектор $\mathbf{t}_k$. Для перехода на следующий шаг необходимо вычесть из матриц $\mathbf{X}_k$ и $\mathbf{Y}_k$ одноранговые аппроксимации $\mathbf{t}_k\mathbf{p}_k^\intercal$ и $\mathbf{t}_k\mathbf{q}_k^\intercal$

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k\mathbf{p}_k^\intercal = \mathbf{X} - \sum_k \mathbf{t}_k\mathbf{p}_k^\intercal,$$

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k - \mathbf{t}_k\mathbf{q}_k^\intercal = \mathbf{Y} - \sum_k \mathbf{t}_k\mathbf{q}_k^\intercal.$$

Тогда каждый следующий вектор $\mathbf{t}_k$ оказывается ортогонален всем векторам $\mathbf{t}_i$, $i = 1, \ldots, k$.

На Рис. 1.1 продемонстрирован результат работы алгоритма PLS для случая, когда размерности пространств объектов, ответов и латентного пространства равны 2 ($n = r = l = 2$). Синими и зелёными точками изображены строки матриц $\mathbf{X}$ и $\mathbf{Y}$. Точки были сгенерированы из нормального распределения с нулевым матожиданием. Красным контуром показаны линии уровня матриц ковариаций распределений. Черным изображены единичные окружности. Красные стрелки соответствуют главным компонентам. Черные стрелки соответствуют

векторам матриц $\mathbf{W}$ и $\mathbf{C}$ алгоритма PLS. Вектора $\mathbf{t}_k$ и $\mathbf{u}_k$ равны проекциям матриц $\mathbf{X}_k$ и $\mathbf{Y}_k$ на вектора $\mathbf{w}_k$ и $\mathbf{c}_k$ соответственно и изображены черными плюсами. Учёт взаимной связи между матрицами $\mathbf{X}$ и $\mathbf{Y}$ отклоняет вектора $\mathbf{w}_k$ и $\mathbf{c}_k$ от направления главных компонент. Вектора $\mathbf{w}_k$ отклоняются незначительно. На первой итерации $\mathbf{c}_1$ близок к pc_1, но вектора $\mathbf{c}_k$, найденные на следующих итерациях могут оказаться сильно коррелированными. Это происходит в следствие того, что из матрицы $\mathbf{Y}$ на каждом шаге вычитается одноранговая аппроксимация, найденная в пространстве матрицы $\mathbf{X}_k$.

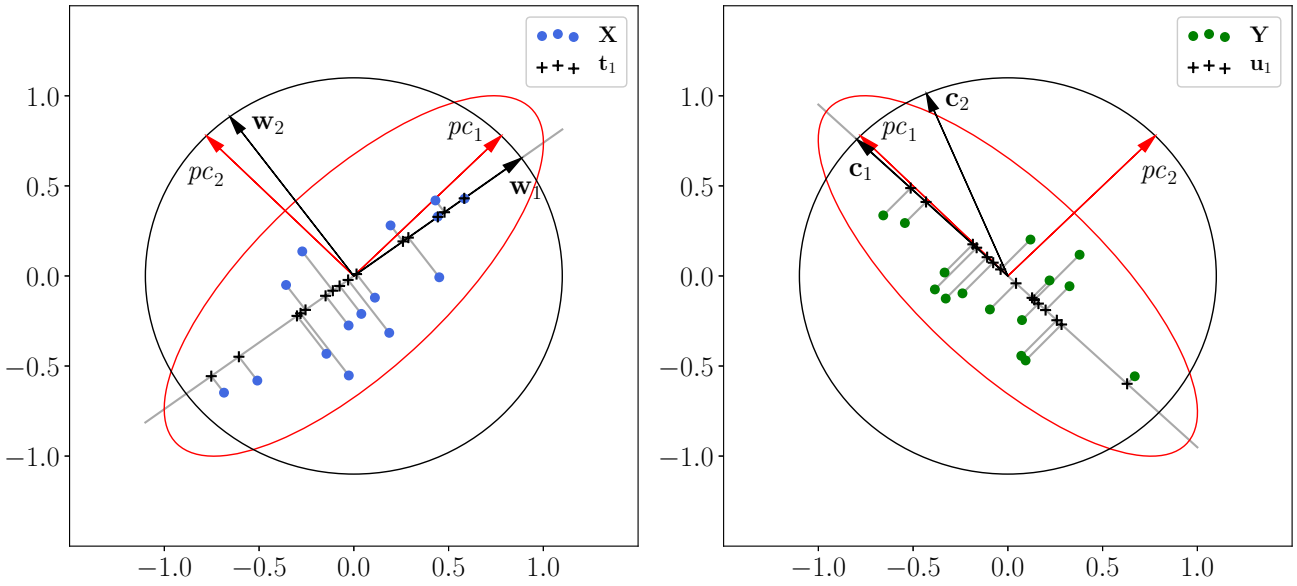

Рис. 1.1. Иллюстрация алгоритма PLS

Для получения прогнозов модели и нахождения параметров модели домножим справа формулу (1.3) на матрицу $\mathbf{W}$. Строки матрицы невязок $\mathbf{E}$ ортогональны столбцам матрицы $\mathbf{W}$, поэтому

$$\mathbf{XW} = \mathbf{TP}^{\intercal}\mathbf{W}.$$

Линейное преобразование между объектами в исходном и латентном пространстве имеет вид

$$\mathbf{T} = \mathbf{XW}^{*}, \tag{1.5}$$

где $\mathbf{W}^{*} = \mathbf{W}(\mathbf{P}^{\intercal}\mathbf{W})^{-1}$.

Матрица параметров модели 1.1 находится из уравнений (1.4), (1.5)

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^{\intercal} + \mathbf{E} = \mathbf{X}\mathbf{W}^{*}\mathbf{Q}^{\intercal} + \mathbf{E} = \mathbf{X}\mathbf{\Theta} + \mathbf{E}.$$

Таким образом, параметры модели (1.1) равны

$$\mathbf{\Theta} = \mathbf{W}(\mathbf{P}^{\intercal}\mathbf{W})^{-1}\mathbf{Q}^{\intercal}. \tag{1.6}$$

Финальная модель (1.3.) является линейной, низкоразмерной в скрытом пространстве. Это снижает избыточность данных и повышает стабильность модели.

Глава 2
Выбор признаков

2.1. Выбор признаков

Задача выбора признаков заключается в поиске оптимального подмножества признаков $\mathcal{A}$ среди всех возможных $2^n - 1$ вариантов Существует взаимооднозначное отображение между подмножеством $\mathcal{A}$ и булевым вектором $\mathbf{a} \in \{0,1\}^n$, компоненты которого указывают, выбран ли признак. Для нахождения оптимального вектора $\mathbf{a}$ введем функцию ошибки выбора признаков $S(\mathbf{a}, \mathbf{X}, \mathbf{Y})$. Проблема выбора признаков принимает вид:

$$\mathbf{a} = \arg\min_{\mathbf{a}' \in \{0,1\}^n} S(\mathbf{a}', \mathbf{X}, \mathbf{Y}). \tag{2.1}$$

Целью выбора признаков является построение функции $S(\mathbf{a}, \mathbf{X}, \mathbf{Y})$. Конкретные примеры данной функции для рассматриваемых алгоритмов выбора признаков приведены ниже и обобщены в таблице 2.1.

Задача (2.1) имеет дискретную область определения $\{0,1\}^n$. Для решения данной задачи применяется релаксация задачи (2.1) к непрерывной области определения $[0,1]^n$. Релаксированная задача выбора функции имеет следующий вид:

$$\mathbf{z} = \arg\min_{\mathbf{z}' \in [0,1]^n} S(\mathbf{z}', \mathbf{X}, \mathbf{Y}). \tag{2.2}$$

Здесь, компоненты вектора $\mathbf{z}$ – значения нормированных коэффициентов важности признаков. Сначала решается задача (2.2), для получения вектора важности $\mathbf{z}$. Затем решение (2.1) восстанавливается с помощью отсечения по порогу следующим образом:

$$\mathbf{a} = [a_j]_{j=1}^n, \quad a_j = \begin{cases} 1, & z_j > \tau; \\ 0, & \text{в противном случае.} \end{cases}$$

τ – гиперпараметр, который может быть подобран вручную или выбран с помощью кросс-валидации.

Как только решение $\mathbf{a}$ задачи (2.1) получено, задача (1.2) принимает вид:

$$\mathcal{L}(\boldsymbol{\Theta}_{\mathcal{A}}, \mathbf{X}_{\mathcal{A}}, \mathbf{Y}) = \left\| \mathbf{Y} - \mathbf{X}_{\mathcal{A}} \boldsymbol{\Theta}_{\mathcal{A}}^{\intercal} \right\|_2^2 \to \min_{\boldsymbol{\Theta}_{\mathcal{A}}},$$

где индекс $\mathcal{A}$ обозначает подматрицу со столбцами, индексы которых содержатся в $\mathcal{A}$.

2.2. Выбор признаков с помощью квадратичного программирования

Если между столбцами матрицы плана $\mathbf{X}$ существует линейная зависимость, то решение задачи линейной регрессии

$$\|\boldsymbol{\nu} - \mathbf{X}\mathbf{w}\|_2^2 \to \min_{\mathbf{w} \in \mathbb{R}^n}. \tag{2.3}$$

оказывается неустойчивым. Методы выбора признаков находят подмножество $\mathcal{A} \in \{1, \ldots, n\}$ оптимальных столбцов $\mathbf{X}$.

Алгоритм QPFS выбирает некоррелированные признаки, релевантные целевому вектору $\mathbf{y}$. Чтобы формализовать этот подход, введем две функции: $\mathrm{Sim}(\mathbf{X})$ и $\mathrm{Rel}(\mathbf{X}, \mathbf{y})$. $\mathrm{Sim}(\mathbf{X})$ контролирует избыточность между признаками, $\mathrm{Rel}(\mathbf{X}, \mathbf{y})$ содержит релевантности между каждым признаком и целевым вектором. Мы хотим минимизировать функцию Sim и максимизировать Rel одновременно.

QPFS предлагает явный способ построения функций Sim и Rel. Алгоритм минимизирует следующую функцию ошибки

$$\underbrace{\mathbf{a}^{\intercal}\mathbf{Q}\mathbf{a}}_{\mathrm{Sim}} - \alpha \cdot \underbrace{\mathbf{b}^{\intercal}\mathbf{a}}_{\mathrm{Rel}} \to \min_{\substack{\mathbf{a} \in \mathbb{R}_+^n \\ \|\mathbf{a}\|_1 = 1}}. \tag{2.4}$$

Элементы матрицы $\mathbf{Q} \in \mathbb{R}^{n \times n}$ содержат коэффициенты попарного сходства между признаками. Вектор $\mathbf{b} \in \mathbb{R}^n$ выражает сходство между каждым признаком и целевым вектором $\mathbf{y}$. Нормированный вектор $\mathbf{a}$ отражает важность каждого признака. Функция ошибки (2.4) штрафует зависимые признаки функцией

Sim и штрафует признаки, не релевантные к целевой переменной функцией Rel. Параметр α позволяет контролировать компромисс между функциями Sim и Rel. Авторы оригинальной статьи о QPFS предложили способ выбора α, чтобы уравновесить вклад членов $\mathrm{Sim}(\mathbf{X})$ и $\mathrm{Rel}(\mathbf{X}, \mathbf{y})$

$$\alpha = \frac{\overline{\mathbf{Q}}}{\overline{\mathbf{Q}} + \overline{\mathbf{b}}}, \quad \overline{\mathbf{Q}} = \mathrm{mean}(\mathbf{Q}), \;\; \overline{\mathbf{b}} = \mathrm{mean}(\mathbf{b}).$$

Чтобы выделить оптимальное подмножество признаков, применяется отсечение по порогу:

$$j \in \mathcal{A} \Leftrightarrow a_j > \tau.$$

Для измерения сходства используется выборочный коэффициент корреляции Пирсона между парами признаков для функции Sim, и между признаками и целевым вектором для функции Rel:

$$\mathbf{Q} = \left[|\mathrm{corr}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)|\right]_{i,j=1}^{n}, \quad \mathbf{b} = \left[|\mathrm{corr}(\boldsymbol{\chi}_i, \boldsymbol{\nu})|\right]_{i=1}^{n}. \tag{2.5}$$

Здесь

$$\mathrm{corr}(\boldsymbol{\chi}, \boldsymbol{\nu}) = \frac{\sum_{i=1}^{m}(\boldsymbol{\chi}_i - \overline{\boldsymbol{\chi}})(\boldsymbol{\nu}_i - \overline{\boldsymbol{\nu}})}{\sqrt{\sum_{i=1}^{m}(\boldsymbol{\chi}_i - \overline{\boldsymbol{\chi}})^2 \sum_{i=1}^{m}(\boldsymbol{\nu}_i - \overline{\boldsymbol{\nu}})^2}}.$$

Другие способы определения $\mathbf{Q}$ и $\mathbf{b}$ рассматриваются в [6]. В работе [6] показано, что алгоритм QPFS превосходит многие существующие алгоритмы выбора функций на различных критериях качества.

Задача (2.4) является выпуклой, если матрица $\mathbf{Q}$ является неотрицательно определенной. В общем случае это не всегда верно. Чтобы удовлетворить этому условию спектр матрицы $\mathbf{Q}$ смещается, и матрица $\mathbf{Q}$ заменяется на $\mathbf{Q} - \lambda_{\min}\mathbf{I}$, где $\lambda_{\min}$ является минимальным собственным значением $\mathbf{Q}$.

2.3. Многомерный QPFS

Здесь описаны предлагаемые методы выбора признаков для случая нескольких многомерной целевой переменной. Если пространство целевых переменных

многомерно, компоненты целевой переменной могут коррелировать между собой. В этом разделе предлагаются алгоритмы, учитывающие зависимости как во входном, так и в целевом пространствах.

2.3.1. Агрегация релевантностей (RelAgg)

В работе [7], чтобы применить алгоритм QPFS к многомерному случаю ($r > 1$), релевантности признаков агрегируются по всем r компонентам. Член $\mathrm{Sim}(\mathbf{X})$ остаётся без изменений, матрица $\mathbf{Q}$ определяется как (2.5). Вектор $\mathbf{b}$ агрегируется по всем компонентам целевой переменной и определяется как

$$\mathbf{b} = \left[\sum_{k=1}^{r} |\mathrm{corr}(\boldsymbol{\chi}_i, \boldsymbol{\nu}_k)| \right]_{i=1}^{n}.$$

Недостатком такого подхода является отсутствие учёта зависимостей в столбцах матрицы $\mathbf{Y}$. Рассмотрим следующий пример:

$$\mathbf{X} = [\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3], \quad \mathbf{Y} = [\underbrace{\boldsymbol{\nu}_1, \boldsymbol{\nu}_1, \ldots, \boldsymbol{\nu}_1}_{r-1}, \boldsymbol{\nu}_2].$$

Пусть матрица $\mathbf{X}$ содержит 3 столбца, матрица $\mathbf{Y}$ – r столбцов, где первые $r-1$ компонент целевой переменной идентичны. Попарные сходства признаков задаются матрицей $\mathbf{Q}$. Компоненты матрицы $\mathbf{B}$ содержит попарные сходства признаков и целевых столбцов. Вектор $\mathbf{b}$ получен суммированием матрицы $\mathbf{B}$ по столбцами

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.8 \\ 0 & 0.8 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.4 & \ldots & 0.4 & 0 \\ 0.5 & \ldots & 0.5 & 0.8 \\ 0.8 & \ldots & 0.8 & 0.1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} (r-1) \cdot 0.4 + 0 \\ (r-1) \cdot 0.5 + 0.8 \\ (r-1) \cdot 0.8 + 0.1 \end{bmatrix}. \tag{2.6}$$

Пусть необходимо выбрать только 2 признака. В данном случае оптимальным подмножеством признаков является $[\boldsymbol{\chi}_1, \boldsymbol{\chi}_2]$. Признак $\boldsymbol{\chi}_2$ предсказывает второй

целевой столбец $\boldsymbol{\nu}_2$, комбинация признаков $\boldsymbol{\chi}_1, \boldsymbol{\chi}_2$ прогнозирует первый целевой столбец. Алгоритм QPFS для $r = 2$ дает решение $\mathbf{z} = [0.37, 0.61, 0.02]$. Это совпадает с описанным решением. Однако, если добавить коллинеарные столбцы в матрицу $\mathbf{Y}$ и увеличить r до 5, то решением QPFS будет $\mathbf{z} = [0.40, 0.17, 0.43]$. Здесь потерян признак $\boldsymbol{\chi}_2$ и выбран избыточный признак $\boldsymbol{\chi}_3$. В следующих подразделах предлагаются обобщения алгоритма QPFS, которые позволяют бороться с проблемой данного примера.

2.3.2. Симметричный учёт важности (SymImp)

Чтобы учесть зависимости в столбцах матрицы $\mathbf{Y}$, обобщим функцию QPFS (2.4) для многомерного случая ($r > 1$). Добавим член $\mathrm{Sim}(\mathbf{Y})$ и изменим член $\mathrm{Rel}(\mathbf{X}, \mathbf{Y})$ следующим образом:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}} \mathbf{Q}_x \mathbf{z}_x}_{\mathrm{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}} \mathbf{B} \mathbf{z}_y}_{\mathrm{Rel}(\mathbf{X},\mathbf{Y})} + \alpha_3 \cdot \underbrace{\mathbf{z}_y^{\mathsf{T}} \mathbf{Q}_y \mathbf{z}_y}_{\mathrm{Sim}(\mathbf{Y})} \to \min_{\substack{\mathbf{z}_x \geqslant \mathbf{0}_n,\, \mathbf{1}_n^{\mathsf{T}} \mathbf{z}_x = 1 \\ \mathbf{z}_y \geqslant \mathbf{0}_r,\, \mathbf{1}_r^{\mathsf{T}} \mathbf{z}_y = 1}} . \tag{2.7}$$

Определим элементы матриц $\mathbf{Q}_x \in \mathbb{R}^{n \times n}$, $\mathbf{Q}_y \in \mathbb{R}^{r \times r}$ и $\mathbf{B} \in \mathbb{R}^{n \times r}$ следующим образом:

$$\mathbf{Q}_x = \left[|\mathrm{corr}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)| \right]_{i,j=1}^{n}, \quad \mathbf{Q}_y = \left[|\mathrm{corr}(\boldsymbol{\nu}_i, \boldsymbol{\nu}_j)| \right]_{i,j=1}^{r}, \quad \mathbf{B} = \left[|\mathrm{corr}(\boldsymbol{\chi}_i, \boldsymbol{\nu}_j)| \right]_{\substack{i=1,\ldots,n \\ j=1,\ldots,r}}.$$

Вектор $\mathbf{z}_x$ содержит коэффициенты важности признаков, $\mathbf{z}_y$ – коэфициенты важности целевых столбцов. Коррелированные целевые столбцы штрафуются членом $\mathrm{Sim}(\mathbf{Y})$ и получают более низкие значения важности.

Коэффициенты α_1, α_2, и α_3 контролируют влияние каждого члена на функцию (2.7) и удовлетворяют следующим условиям:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1, \quad \alpha_i \geqslant 0,\ i = 1, 2, 3.$$

Утверждение 4. Баланс между $\mathrm{Sim}(\mathbf{X})$, $\mathrm{Rel}(\mathbf{X}, \mathbf{Y})$ и $\mathrm{Sim}(\mathbf{Y})$ в задаче (2.7) достигается при:

$$\alpha_1 \propto \overline{\mathbf{Q}}_y \overline{\mathbf{B}}; \quad \alpha_2 \propto \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y; \quad \alpha_3 \propto \overline{\mathbf{Q}}_x \overline{\mathbf{B}}. \tag{2.8}$$

Доказательство. Значения α_1, α_2, и α_3 получаются путем решения следующих уравнений:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1;$$
$$\alpha_1 \overline{\mathbf{Q}}_x = \alpha_2 \overline{\mathbf{B}} = \alpha_3 \overline{\mathbf{Q}}_y.$$

Здесь средние значения $\overline{\mathbf{Q}}_x$, $\overline{\mathbf{B}}$ и $\overline{\mathbf{Q}}_y$ соответствующих матриц $\mathbf{Q}_x$, $\mathbf{B}$ и $\mathbf{Q}_y$ - средние значения членов $\mathrm{Sim}(\mathbf{X})$, $\mathrm{Rel}(\mathbf{X}, \mathbf{Y})$ и $\mathrm{Sim}(\mathbf{Y})$. □

Для изучения зависимости $\mathrm{Sim}(\mathbf{Y})$ на функцию (2.7), зафиксируем соотношение между α_1 и α_2:

$$\alpha_1 = \frac{(1-\alpha_3)\overline{\mathbf{B}}}{\overline{\mathbf{Q}}_x + \overline{\mathbf{B}}}; \quad \alpha_2 = \frac{(1-\alpha_3)\overline{\mathbf{Q}}_x}{\overline{\mathbf{Q}}_x + \overline{\mathbf{B}}}; \quad \alpha_3 \in [0, 1]. \tag{2.9}$$

Применим предложенный алгоритм к приведенному примеру (2.6). Матрица $\mathbf{Q}$ соответствует матрице $\mathbf{Q}_x$. Определим матрицы $\mathbf{Q}_y$ как $\mathrm{corr}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2) = 0.2$, а все остальные элементы зададим 1. Рисунок 2.1 показывает значение векторов важности признаков $\mathbf{z}_x$ и целевых векторов $\mathbf{z}_y$ в зависимости от значения коэффициента α_3. Если α_3 мало, важность всех целевых векторов практически идентична и важность признака $\boldsymbol{\chi}_3$ выше важности признака $\boldsymbol{\chi}_2$. При увеличении α_3 д 0.2, коэффициент важности $\mathbf{z}_{y,5}$ целевого вектора $\boldsymbol{\nu}_5$ увеличивается наряду с важностью признака $\boldsymbol{\chi}_2$.

2.3.3. Минимакс QPFS (MinMax)

Функция (2.7) является симметричной по отношению к $\mathbf{z}_x$ и $\mathbf{z}_y$. Она штрафует признаки, которые коррелированы и не имеют отношения к целевым векторам. Кроме того, она штрафует цели, которые коррелированы между собой и недостаточно коррелируют с признаками. Это приводит к малым значениям важности для целевых векторов, которые слабо коррелируют с признаками, и большим значениям для целевых векторов, которые сильно коррелируют с признаков. Этот результат противоречит интуиции. Цель –предсказать все целевые

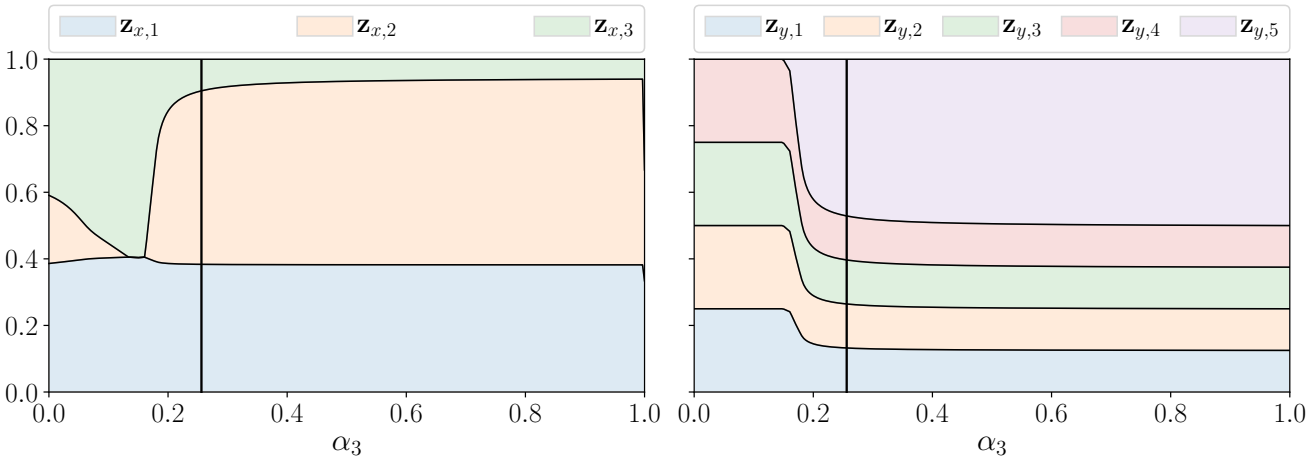

Рис. 2.1. Важности признаков $\mathbf{z}_x$ и целевых векторов $\mathbf{z}_y$ в зависимости от α_3 для рассмотренного примера

вектора, особенно те, которые слабо коррелируют с признаками, по релевантным и некоррелированным признакам. Данная цель выражается в виде двух взаимосвязанных задач:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\mathrm{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\mathrm{Rel}(\mathbf{X},\mathbf{Y})} \to \min_{\substack{\mathbf{z}_x \geqslant \mathbf{0}_n, \\ \mathbf{1}_n^\top \mathbf{z}_x = 1}}; \tag{2.10}$$

$$\alpha_3 \cdot \underbrace{\mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\mathrm{Sim}(\mathbf{Y})} + \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\mathrm{Rel}(\mathbf{X},\mathbf{Y})} \to \min_{\substack{\mathbf{z}_y \geqslant \mathbf{0}_r, \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}}. \tag{2.11}$$

Разница между (2.10) и (2.11) является знак перед членом Rel. В пространстве входных объектов нерелевантные компоненты должны иметь меньшие значения важности. В то же время целевые вектора, не релевантные признакам, должны иметь большую важность. Задачи (2.10) и (2.11) объединяются в совместную минимакс или максмин постановку

$$\min_{\substack{\mathbf{z}_x \geqslant \mathbf{0}_n \\ \mathbf{1}_n^\top \mathbf{z}_x = 1}} \max_{\substack{\mathbf{z}_y \geqslant \mathbf{0}_r \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}} f(\mathbf{z}_x, \mathbf{z}_y), \quad \left(\text{или} \max_{\substack{\mathbf{z}_y \geqslant \mathbf{0}_r \\ \mathbf{1}_r^\top \mathbf{z}_y = 1}} \min_{\substack{\mathbf{z}_x \geqslant \mathbf{0}_n \\ \mathbf{1}_n^\top \mathbf{z}_x = 1}} f(\mathbf{z}_x, \mathbf{z}_y) \right), \tag{2.12}$$

где

$$f(\mathbf{z}_x, \mathbf{z}_y) = \alpha_1 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{Q}_x \mathbf{z}_x}_{\mathrm{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^\top \mathbf{B} \mathbf{z}_y}_{\mathrm{Rel}(\mathbf{X},\mathbf{Y})} - \alpha_3 \cdot \underbrace{\mathbf{z}_y^\top \mathbf{Q}_y \mathbf{z}_y}_{\mathrm{Sim}(\mathbf{Y})}.$$

Теорема 1. Для положительно определенной матрицы $\mathbf{Q}_x$ и $\mathbf{Q}_y$, максмин и минимакс задачи (2.12) имеют одинаковое оптимальное значение.

Доказательство. Введём обозначение

$$\mathbb{C}^n = \{\mathbf{z} : \mathbf{z} \geqslant \mathbf{0}_n,\ \mathbf{1}_n^\intercal \mathbf{z} = 1\}, \quad \mathbb{C}^r = \{\mathbf{z} : \mathbf{z} \geqslant \mathbf{0}_r,\ \mathbf{1}_r^\intercal \mathbf{z} = 1\}.$$

Множества $\mathbb{C}^n$ и ^{r} - компактные и выпуклые. Функция $f : \mathbb{C}^n \times \mathbb{C}^r \to \mathbb{R}$ является непрерывной. Если $\mathbf{Q}_x$ и $\mathbf{Q}_y$ являются положительно определенными матрицами, функция f выпукло-вогнутая. Т. е., $f(\cdot, \mathbf{z}_y) : \mathbb{C}^n \to \mathbb{R}$ выпуклая при фиксированном $\mathbf{z}_y$, а $f(\mathbf{z}_x, \cdot) : \mathbb{C}^r \to \mathbb{R}$ вогнута при фиксированном $\mathbf{z}_x$. В этом случае по теореме Неймана о минимаксе

$$\min_{\mathbf{z}_x \in \mathbb{C}^n} \max_{\mathbf{z}_y \in \mathbb{C}^r} f(\mathbf{z}_x, \mathbf{z}_y) = \max_{\mathbf{z}_y \in \mathbb{C}^r} \min_{\mathbf{z}_x \in \mathbb{C}^n} f(\mathbf{z}_x, \mathbf{z}_y).$$

□

Для решения минимакс задачи (2.12), зафиксируем некоторый $\mathbf{z}_x \in {}^n$. Для фиксированного вектора $\mathbf{z}_x$ решаем задачу

$$\max_{\mathbf{z}_y \in \mathbb{C}_r} f(\mathbf{z}_x, \mathbf{z}_y) = \max_{\substack{\mathbf{z}_y \geqslant \mathbf{0}_r \\ \mathbf{1}_r^\intercal \mathbf{z}_y = 1}} \left[\alpha_1 \cdot \mathbf{z}_x^\intercal \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\intercal \mathbf{B} \mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^\intercal \mathbf{Q}_y \mathbf{z}_y\right]. \tag{2.13}$$

Лагранжиан данной задачи:

$$L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) = \alpha_1 \cdot \mathbf{z}_x^\intercal \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\intercal \mathbf{B} \mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^\intercal \mathbf{Q}_y \mathbf{z}_y + \lambda \cdot (\mathbf{1}_r^\intercal \mathbf{z}_y - 1) + \boldsymbol{\mu}^\intercal \mathbf{z}_y.$$

Здесь вектор множителей Лагранжа $\boldsymbol{\mu}$, который соответствует ограничениям на неравенства $\mathbf{z}_y \geqslant \mathbf{0}_r$, является неотрицательным. Двойственной задачей является

$$\min_{\lambda,\, \boldsymbol{\mu} \geqslant \mathbf{0}_r} g(\mathbf{z}_x, \lambda, \boldsymbol{\mu}) = \min_{\lambda,\, \boldsymbol{\mu} \geqslant \mathbf{0}_r} \left[\max_{\mathbf{z}_y \in \mathbb{R}^r} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu})\right]. \tag{2.14}$$

Для задачи квадратичного программирования (2.13) с положительно определенными матрицами $\mathbf{Q}_x$ и $\mathbf{Q}_y$ выполняются условия сильной двойственности. Таким образом, оптимальное значение (2.13) равно оптимальному значению (2.14). Это позволяет перейти от решения задачи (2.12) к решению задачи

$$\min_{\mathbf{z}_x \in \mathbb{C}^n,\, \lambda,\, \boldsymbol{\mu} \geqslant \mathbf{0}_r} g(\mathbf{z}_y, \lambda, \boldsymbol{\mu}). \tag{2.15}$$

Полагая градиент $\nabla_{\mathbf{z}_y} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu})$ равным нулю, получим оптимальное значение $\mathbf{z}_y$:

$$\mathbf{z}_y = \frac{1}{2\alpha_3}\mathbf{Q}_y^{-1}\left(-\alpha_2 \cdot \mathbf{B}^\intercal \mathbf{z}_x + \lambda \cdot \mathbf{1}_r + \boldsymbol{\mu}\right). \tag{2.16}$$

Двойственная функция принимает вид

$$\begin{aligned} g(\mathbf{z}_x, \lambda, \boldsymbol{\mu}) = \max_{\mathbf{z}_y \in \mathbb{R}^r} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) = \mathbf{z}_x^\intercal \left(-\frac{\alpha_2^2}{4\alpha_3} \cdot \mathbf{B}\mathbf{Q}_y^{-1}\mathbf{B}^\intercal - \alpha_1 \cdot \mathbf{Q}_x\right)\mathbf{z}_x \\ - \frac{1}{4\alpha_3}\lambda^2 \cdot \mathbf{1}_r^\intercal \mathbf{Q}_y^{-1}\mathbf{1}_r - \frac{1}{4\alpha_3} \cdot \boldsymbol{\mu}^\intercal \mathbf{Q}_y^{-1}\boldsymbol{\mu} + \frac{\alpha_2}{2\alpha_3}\lambda \cdot \mathbf{1}_r^\intercal \mathbf{Q}_y^{-1}\mathbf{B}^\intercal \mathbf{z}_x \\ - \frac{1}{2\alpha_3}\lambda \cdot \mathbf{1}_r^\intercal \mathbf{Q}_y^{-1}\boldsymbol{\mu} + \frac{\alpha_2}{2\alpha_3} \cdot \boldsymbol{\mu}^\intercal \mathbf{Q}_y^{-1}\mathbf{B}^\intercal \mathbf{z}_x + \lambda. \end{aligned} \tag{2.17}$$

Тем самым задача (2.15) является квадратичной задачей с $n + r + 1$ переменными.

2.3.4. Несимметричный учёт важности (SymImp)

Естественным способом преодоления проблемы алгоритма SymImp является добавление штрафа для целевых векторов, которые коррелируют с признаками. Добавим линейный член $\mathbf{b}^\intercal \mathbf{z}_y$ в член $\mathrm{Rel}(\mathbf{X}, \mathbf{Y})$ следующим образом:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^\intercal \mathbf{Q}_x \mathbf{z}_x}_{\mathrm{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\left(\mathbf{z}_x^\intercal \mathbf{B}\mathbf{z}_y - \mathbf{b}^\intercal \mathbf{z}_y\right)}_{\mathrm{Rel}(\mathbf{X},\mathbf{Y})} + \alpha_3 \cdot \underbrace{\mathbf{z}_y^\intercal \mathbf{Q}_y \mathbf{z}_y}_{\mathrm{Sim}(\mathbf{Y})} \to \min_{\substack{\mathbf{z}_x \geqslant \mathbf{0}_n,\, \mathbf{1}_n^\intercal \mathbf{z}_x = 1 \\ \mathbf{z}_y \geqslant \mathbf{0}_r,\, \mathbf{1}_r^\intercal \mathbf{z}_y = 1}}. \tag{2.18}$$

Утверждение 5. Пусть вектор $\mathbf{b}$ равен

$$b_j = \max_{i=1,\ldots n} [\mathbf{B}]_{i,j}.$$

Тогда значение коэффициентов важности вектора $\mathbf{z}_y$ будут неотрицательными в $\mathrm{Rel}(\mathbf{X}, \mathbf{Y})$ для задачи (2.18).

Доказательство. Утверждение следует из факта

$$\sum_{i=1}^{n} z_i b_{ij} \leqslant \left(\sum_{i=1}^{n} z_i\right) \max_{i=1,\ldots n} b_{ij} = \max_{i=1,\ldots n} b_{ij},$$

где $z_i \geqslant 0$ и $\sum_{i=1}^{n} z_i = 1$. □

Следовательно, функция (2.18) штрафует в меньшей мере признаки, которые имеют отношение к целевым векторам, и целевые вектора, которые недостаточно коррелированы с признаками.

Утверждение 6. Баланс между членами $\mathrm{Sim}(\mathbf{X})$, $\mathrm{Rel}(\mathbf{X},\mathbf{Y})$ и $\mathrm{Rel}(\mathbf{X},\mathbf{Y})$ для задачи (2.18) достигается при следующих коэффициентах:

$$\alpha_1 \propto \overline{\mathbf{Q}}_y \left(\overline{\mathbf{b}} - \overline{\mathbf{B}}\right); \quad \alpha_2 \propto \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y; \quad \alpha_3 \propto \overline{\mathbf{Q}}_x \overline{\mathbf{B}}.$$

Доказательство. Необходимые значения α_1, α_2, и α_3 являются решением следующей системы уравнений:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1; \tag{2.19}$$

$$\alpha_1 \overline{\mathbf{Q}}_x = \alpha_2 \overline{\mathbf{B}}; \tag{2.20}$$

$$\alpha_2 \left(\overline{\mathbf{b}} - \overline{\mathbf{B}}\right) = \alpha_3 \overline{\mathbf{Q}}_y. \tag{2.21}$$

Здесь, в (2.20) уравновешены $\mathrm{Sim}(\mathbf{X})$ с первым слагаемым $\mathrm{Rel}(\mathbf{X},\mathbf{Y})$, а в (2.21) уравновешены $\mathrm{Sim}(\mathbf{Y})$ с $\mathrm{Rel}(\mathbf{X},\mathbf{Y})$. □

Утверждение 7. Для случая $r = 1$, предложенные функции (2.7), (2.12) и (2.18) совпадают с оригинальным алгоритмом QPFS (2.4).

Доказательство. Если r равно 1, то $\mathbf{Q}_y = q_y$ - скаляр, $\mathbf{z}_y = 1$ и $\mathbf{B} = \mathbf{b}$. Задачи (2.7), (2.12) и (2.18) принимают вид

$$\alpha_1 \cdot \mathbf{z}_x^\intercal \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^\intercal \mathbf{b} \to \min_{\mathbf{z}_x \geqslant \mathbf{0}_n,\, \mathbf{1}_n^\intercal \mathbf{z}_x = 1}.$$

При $\alpha = \frac{\alpha_2}{\alpha_1 + \alpha_2}$ последняя задачи принимает вид (2.4). □

Таблица 2.1 демонстрирует основные идеи и функции ошибок для каждого алгоритма. RelAgg является базовой стратегией и не учитывает корреляции в целевом пространстве. SymImp штрафует попарные корреляции между целевыми векторами. MinMax более чувствителен к целевым векторам, которые трудно предсказать. Стратегия Asymimp добавляет линейный член к функции SymImp, чтобы сделать вклад признаков и целевых векторов асимметричным.

Таблица 2.1

Обзор предлагаемых обобщений многомерного QPFS алгоритма

Алгоритм	Идея	Функция ошибки $S(\mathbf{z}\|\mathbf{X},\mathbf{Y})$
RelAgg	$\min\big[\mathrm{Sim}(\mathbf{X}) - \mathrm{Rel}(\mathbf{X},\mathbf{Y})\big]$	$\min\limits_{\mathbf{z}_x}\big[(1-\alpha)\cdot\mathbf{z}_x^\intercal\mathbf{Q}_x\mathbf{z}_x - \alpha\cdot\mathbf{z}_x^\intercal\mathbf{B}\mathbf{1}_r\big]$
SymImp	$\min\big[\mathrm{Sim}(\mathbf{X}) - \mathrm{Rel}(\mathbf{X},\mathbf{Y}) + \mathrm{Sim}(\mathbf{Y})\big]$	$\min\limits_{\mathbf{z}_x,\,\mathbf{z}_y}\Big[\alpha_1\cdot\mathbf{z}_x^\intercal\mathbf{Q}_x\mathbf{z}_x - \alpha_2\cdot\mathbf{z}_x^\intercal\mathbf{B}\mathbf{z}_y + \alpha_3\cdot\mathbf{z}_y^\intercal\mathbf{Q}_y\mathbf{z}_y\Big]$
MinMax	$\min\big[\mathrm{Sim}(\mathbf{X}) - \mathrm{Rel}(\mathbf{X},\mathbf{Y})\big]$ $\max\big[\mathrm{Rel}(\mathbf{X},\mathbf{Y}) + \mathrm{Sim}(\mathbf{Y})\big]$	$\min\limits_{\mathbf{z}_x}\max\limits_{\mathbf{z}_y}\big[\alpha_1\cdot\mathbf{z}_x^\intercal\mathbf{Q}_x\mathbf{z}_x - \alpha_2\cdot\mathbf{z}_x^\intercal\mathbf{B}\mathbf{z}_y - \alpha_3\cdot\mathbf{z}_y^\intercal\mathbf{Q}_y\mathbf{z}_y\big]$
AsymImp	$\min\big[\mathrm{Sim}(\mathbf{X}) - \mathrm{Rel}(\mathbf{X},\mathbf{Y})\big]$ $\max\big[\mathrm{Rel}(\mathbf{X},\mathbf{Y}) + \mathrm{Sim}(\mathbf{Y})\big]$	$\min\limits_{\mathbf{z}_x,\mathbf{z}_y}\big[\alpha_1\cdot\mathbf{z}_x^\intercal\mathbf{Q}_x\mathbf{z}_x - \alpha_2\cdot\Big(\mathbf{z}_x^\intercal\mathbf{B}\mathbf{z}_y - \mathbf{b}^\intercal\mathbf{z}_y\Big) + \alpha_3\cdot\mathbf{z}_y^\intercal\mathbf{Q}_y\mathbf{z}_y\big]$

Глава 3
Выбор параметров нелинейных моделей

3.1. Задача оптимизации

Модель $f(\mathbf{x},\mathbf{w})$ с параметрами $\mathbf{w}\in\mathbb{R}^p$ предсказывает целевую переменную $y\in\mathbb{Y}$ по объекту $\mathbf{x}\in\mathbb{R}^n$. Пространство $\mathbb{Y}$ представляет собой бинарные метки классов $\{0,1\}$ для задачи двухклассовой классификации и $\mathbb{R}$ для задачи регрессии. Даны матрица плана $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_m]^\intercal \in \mathbb{R}^{m\times n}$ и целевой вектор $\mathbf{y} = [y_1,\ldots,y_m]^\intercal \in \mathbb{Y}^m$. Цель состоит в нахождении оптимальных параметров $\mathbf{w}^*$. Параметры $\mathbf{w}$ вычисляются минимизацией функции ошибки:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathbb{R}^p} \mathcal{L}(\mathbf{w},\mathbf{X},\mathbf{y},f). \tag{3.1}$$

В качестве функции ошибки $\mathcal{L}(\mathbf{w},\mathbf{X},\mathbf{y},f)$ рассматриваются квадратичная ошибка для задачи регрессии:

$$\mathcal{L}(\mathbf{w},\mathbf{X},\mathbf{y},f) = \frac{1}{2}\|\mathbf{y}-\mathbf{f}(\mathbf{X},\mathbf{w})\|_2^2 = \frac{1}{2}\sum_{i=1}^{m}\big(y_i - f(\mathbf{x}_i,\mathbf{w})\big)^2, \tag{3.2}$$

и функция кросс-энтропии для задачи бинарной классификации:

$$\mathcal{L}(\mathbf{w}, \mathbf{X}, \mathbf{y}, f) = \sum_{i=1}^{m} \big[y_i \log f(\mathbf{x}_i, \mathbf{w}) + (1 - y_i) \log\big(1 - f(\mathbf{x}_i, \mathbf{w})\big) \big]. \tag{3.3}$$

Задача (3.1) решается с помощью итеративной процедуры оптимизации. Для получения параметров на шаге k текущие параметры $\mathbf{w}^{k-1}$ обновляются по следующему правилу:

$$\mathbf{w}^k = \mathbf{w}^{k-1} + \Delta\mathbf{w}^{k-1}. \tag{3.4}$$

Авторы используют метод оптимизации Ньютона для выбора вектора обновлений $\Delta\mathbf{w}$.

Метод Ньютона нестабилен и вычислительно сложен. В данной статье предлагается стабильный алгоритм Ньютона. Перед шагом градиента предлагается выбрать подмножество активных параметров модели, которые оказывают наибольшее влияние на функцию ошибки $\mathcal{L}(\mathbf{w}, \mathbf{X}, \mathbf{y}, f)$. Обновление параметров производится только для отобранного множества индексов $\mathcal{A} = \{j : a_j = 1, \mathbf{a} \in \{0,1\}^p\}$

$$\mathbf{w}^k_{\mathcal{A}} = \mathbf{w}^{k-1}_{\mathcal{A}} + \Delta\mathbf{w}^{k-1}_{\mathcal{A}}, \quad \mathbf{w}_{\mathcal{A}} = \{w_j : j \in \mathcal{A}\};$$
$$\mathbf{w}^k_{\bar{\mathcal{A}}} = \mathbf{w}^{k-1}_{\bar{\mathcal{A}}}, \quad \mathbf{w}_{\bar{\mathcal{A}}} = \{w_j : j \notin \mathcal{A}\}.$$

Чтобы выбрать оптимальное подмножество индексов $\mathcal{A}$, из всех возможных $2^p - 1$ подмножеств, вводится функция ошибки

$$\mathbf{a} = \underset{\mathbf{a}' \in \{0,1\}^p}{\arg\min}\, S(\mathbf{a}', \mathbf{X}, \mathbf{y}, f, \mathbf{w}), \tag{3.5}$$

аналогичная функции ошибки (2.1) для задачи выбора признаков. Задача (3.5) решается на каждом шаге k процесса оптимизации для текущих параметров $\mathbf{w}^k$.

Алгоритм QPFS используется для решения задачи (3.5). QPFS выбирает подмножество параметров $\mathbf{a}$ для вектора обновлений $\Delta\mathbf{w}$, которые оказывают наибольшее влияние на вектор остатков и являются попарно независимыми.

Функция ошибки (2.4) соответствует функции ошибки $S(\mathbf{a}, \mathbf{X}, \mathbf{y}, f, \mathbf{w})$

$$\mathbf{a} = \underset{\mathbf{a}' \in \{1,0\}^p}{\arg\max}\, S(\mathbf{a}', \mathbf{X}, \mathbf{y}, f, \mathbf{w}) \Leftrightarrow \underset{\mathbf{a} \in \mathbb{R}^p_+,\, \|\mathbf{a}\|_1 = 1}{\arg\min} \left[\mathbf{a}^\intercal \mathbf{Q} \mathbf{a} - \alpha \cdot \mathbf{b}^\intercal \mathbf{a}\right]. \tag{3.6}$$

В работе показано, что для модели нелинейной регрессии с квадратичной функцией ошибки (3.2) и для модели логистической регрессии с кросс-энтропией (3.3), каждый шаг оптимизации эквивалентен задаче линейной регрессии (2.3).

3.2. Метод Ньютона

Метод Ньютона использует условие оптимизации первого порядка для задачи (3.1) и линеаризует градиент $S(\mathbf{w})$

$$\nabla S(\mathbf{w} + \Delta \mathbf{w}) = \nabla S(\mathbf{w}) + \mathbf{H} \cdot \Delta \mathbf{w} = 0,$$

$$\Delta \mathbf{w} = -\mathbf{H}^{-1} \nabla S(\mathbf{w}).$$

где $\mathbf{H} = \nabla^2 S(\mathbf{w})$ является Гессианом матрицы функции ошибки $S(\mathbf{w})$.

Итерация (3.4) метода Ньютона –

$$\mathbf{w}^k = \mathbf{w}^{k-1} - \mathbf{H}^{-1} \nabla S(\mathbf{w}).$$

Каждая итерация инвертирует матрицу Гессиана. Мерой плохой обусловленности для матрицы Гессиана $\mathbf{H}$ является число обусловленности

$$\varkappa(\mathbf{H}) = \frac{\lambda_{\max}(\mathbf{H})}{\lambda_{\min}(\mathbf{H})},$$

где $\lambda_{\max}(\mathbf{H}), \lambda_{\min}(\mathbf{H})$ являются максимальным и минимальным собственными значениями $\mathbf{H}$. Большое число обусловленности $\varkappa(\mathbf{H})$ приводит к нестабильности процесса оптимизации. Предложенный алгоритм уменьшает размер матрицы Гессиана $\mathbf{H}$. В наших экспериментах это приводит к меньшему числу обусловленности $\varkappa(\mathbf{H})$.

Размер шага метода Ньютона может быть чрезмерно большим. Для управления размером шага обновлений добавим параметр η в правило обновления (3.4)

$$\mathbf{w}^k = \mathbf{w}^{k-1} + \eta\Delta\mathbf{w}^{k-1}, \quad \eta \in [0, 1].$$

Для выбора соответствующего размера шага η используется правило Армихо. Выбирается максимальное η так, чтобы выполнялось следующее условие

$$S(\mathbf{w}^{k-1} + \eta\Delta\mathbf{w}^{k-1}) < S(\mathbf{w}^{k-1}) + \gamma\eta\nabla S^{\intercal}(\mathbf{w}^{k-1})\mathbf{w}^{k-1}, \quad \gamma \in [0, 0.5].$$

3.3. Модели нелинейной и логистической регрессии

3.3.1. Модель нелинейной регрессии

Предположим, что модель $f(\mathbf{x}, \mathbf{w})$ близка к линейной в окрестности точки $\mathbf{w} + \Delta\mathbf{w}$

$$\mathbf{f}(\mathbf{X}, \mathbf{w} + \Delta\mathbf{w}) \approx \mathbf{f}(\mathbf{X}, \mathbf{w}) + \mathbf{J} \cdot \Delta\mathbf{w},$$

где $\mathbf{J} \in \mathbb{R}^{m\times p}$ является матрицы Якоби

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f(\mathbf{x}_1,\mathbf{w})}{\partial w_1} & \dots & \frac{\partial f(\mathbf{x}_1,\mathbf{w})}{\partial w_p} \\ \dots & \dots & \dots \\ \frac{\partial f(\mathbf{x}_m,\mathbf{w})}{\partial w_1} & \dots & \frac{\partial f(\mathbf{x}_m,\mathbf{w})}{\partial w_p} \end{pmatrix}. \tag{3.7}$$

В соответствии с этим предположением градиент $\nabla S(\mathbf{w})$ и Гессиан матрицы $\mathbf{H}$ функции ошибки (3.2) равняются

$$\nabla S(\mathbf{w}) = \mathbf{J}^{\intercal}(\mathbf{y} - \mathbf{f}), \quad \mathbf{H} = \mathbf{J}^{\intercal}\mathbf{J}. \tag{3.8}$$

Это приводит к методу Гаусса-Ньютона и правилу обновления (3.4)

$$\mathbf{w}^k = \mathbf{w}^{k-1} + (\mathbf{J}^{\intercal}\mathbf{J})^{-1}\mathbf{J}^{\intercal}(\mathbf{f} - \mathbf{y}).$$

Вектор обновления $\Delta\mathbf{w}$ является решением задачи линейной регрессии

$$\|\mathbf{z} - \mathbf{F}\Delta\mathbf{w}\|_2^2 \to \min_{\Delta\mathbf{w}\in\mathbb{R}^p}, \tag{3.9}$$

где $\mathbf{z} = \mathbf{f} - \mathbf{y}$ и $\mathbf{F} = \mathbf{J}$.

В качестве нелинейной модели рассматривается модель двухслойной нейронной сеть. В этом случае модель $f(\mathbf{x}, \mathbf{w})$ задается следующим образом:

$$f(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{x}^\intercal \mathbf{W}_1)\mathbf{w}_2.$$

Здесь $\mathbf{W}_1 \in \mathbb{R}^{N \times h}$ – это матрица весов, которые соединяют исходные признаки с h скрытыми нейронами. Функция нелинейности $\sigma(\cdot)$ применяется поэлементно. Веса $\mathbf{w}_2 \in \mathbb{R}^{h \times 1}$ соединяют скрытые нейроны с выходом. Вектор параметров модели $\mathbf{w}$ представляет собой объединение векторизованных матриц $\mathbf{W}_1$, $\mathbf{w}_2$.

3.3.2. Модель логистической регрессии

Для логистической регрессии модель имеет вид $f(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{x}^\intercal \mathbf{w})$ с сигмоидной функцией активации $\sigma(\cdot)$. Градиент и Гессиан функции ошибки (3.3) равны

$$\nabla S(\mathbf{w}) = \mathbf{X}^\intercal(\mathbf{f} - \mathbf{y}), \quad \mathbf{H} = \mathbf{X}^\intercal \mathbf{R} \mathbf{X}, \tag{3.10}$$

где $\mathbf{R}$ – это диагональная матрица с диагональными элементами $f(\mathbf{x}_i, \mathbf{w}) \cdot (1 - f(\mathbf{x}_i, \mathbf{w}))$.

Правило обновления (3.4) в этом случае

$$\mathbf{w}^k = \mathbf{w}^{k-1} + (\mathbf{X}^\intercal \mathbf{R} \mathbf{X})^{-1} \mathbf{X}^\intercal (\mathbf{y} - \mathbf{f}).$$

Этот алгоритм известен как итеративный алгоритм взвешенных наименьших квадратов (IRLS). Вектор обновлений $\Delta\mathbf{w}$ является решением задачи линейной регрессии

$$\|\mathbf{z} - \mathbf{F}\Delta\mathbf{w}\|_2^2 \to \min_{\Delta\mathbf{w} \in \mathbb{R}^p}, \tag{3.11}$$

где $\mathbf{z} = \mathbf{R}^{-1/2}(\mathbf{y} - \mathbf{f})$ и $\mathbf{F} = \mathbf{R}^{1/2}\mathbf{X}$.

3.4. Алгоритм QPFS+Ньютон

Предлагается реализовать алгоритм QPFS для решения задач (3.9) и (3.11). QPFS матрица $\mathbf{Q}$ и вектор $\mathbf{b}$ имеют вид

$$\mathbf{Q} = \mathrm{Sim}(\mathbf{F}), \quad \mathbf{b} = \mathrm{Rel}(\mathbf{F}, \mathbf{z}).$$

Выборочный коэффициент корреляции равен нулю для ортогональных векторов. Покажем, что в оптимальной точке $\mathbf{w}^*$ вектор $\mathbf{z}$ ортогонален столбцам матрицы $\mathbf{F}$. В этом случае вектор $\mathbf{b} = \mathrm{Rel}(\mathbf{F}, \mathbf{z})$ равен нулю. Это означает, что член, учитывающий релевантность, в данном случае исключается. Условие оптимизации первого порядка гарантирует это свойство для модели нелинейной регрессии

$$\mathbf{F}^\intercal \mathbf{z} = \mathbf{J}^\intercal(\mathbf{f} - \mathbf{y}) = -\nabla S(\mathbf{w}^*) = \mathbf{0},$$

и для модели логистической регрессии

$$\mathbf{F}^\intercal \mathbf{z} = \mathbf{X}\mathbf{R}^{-1/2}\mathbf{R}^{1/2}(\mathbf{y} - \mathbf{f}) = \mathbf{X}^\intercal(\mathbf{y} - \mathbf{f}) = \nabla S(\mathbf{w}^*) = \mathbf{0}.$$

Псевдокод предлагаемого алгоритма приведён в алгоритме 2.

Algorithm 2 QPFS + Ньютон алгоритм

Вход: ε – допустимое отклонение;

τ – пороговое значение;

γ – параметр правила Армихо.

Выход: $\mathbf{w}^*$;

инициализировать $\mathbf{w}^0$;

$k := 1$;

повторять

вычислить $\mathbf{z}$ и $\mathbf{F}$ для (3.9) или (3.11) ;

$\mathbf{Q} := \mathrm{Sim}(\mathbf{F})$, $\mathbf{b} := \mathrm{Rel}(\mathbf{F}, \mathbf{z})$, $\alpha = \frac{\overline{\mathbf{Q}}}{\overline{\mathbf{Q}}+\overline{\mathbf{b}}}$;

$\mathbf{a} := \underset{\mathbf{a}\geqslant 0,\, \|\mathbf{a}\|_1=1}{\arg\min}\ \mathbf{a}^\intercal \mathbf{Q}\mathbf{a} - \alpha \cdot \mathbf{b}^\intercal \mathbf{a}$;

$\mathcal{A} := \{j : a_j = 1\}$;

вычислить $\nabla S(\mathbf{w}^{k-1})$, $\mathbf{H}$ для (3.8) или (3.10);

$\Delta \mathbf{w}^{k-1} = -\mathbf{H}^{-1} \nabla S(\mathbf{w}^{k-1})$;

$\eta := \mathrm{ArmijoRule}(\mathbf{w}^{k-1}, \gamma)$;

$\mathbf{w}^k_{\mathcal{A}} = \mathbf{w}^{k-1}_{\mathcal{A}} + \eta \Delta \mathbf{w}^{k-1}_{\mathcal{A}}$;

$k := k + 1$;

пока $\frac{\|\mathbf{w}^k - \mathbf{w}^{k-1}\|}{\|\mathbf{w}^k\|} < \varepsilon$

Глава 4
Метрические методы

связать все определения

4.1. Постановка задачи метрического обучения

Пусть $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{T \times N}$ — множество объектов. Объект $\mathbf{x}_i = [x_i^1, \ldots, x_i^T]^\top$ задан в виде вектора в пространстве признаков. Требуется выявить кластерную структуру данных и разбить множество объектов $\mathbf{X}$ на множество непересекающихся кластеров, т. е. построить отображение

$$a : \mathbf{X} \to \{1, \ldots, K\}.$$

Обозначим $y_i = a(\mathbf{x}_i)$, $y_i \in \{1, \ldots, K\}$, — метка кластера объекта $\mathbf{x}_i$. Необходимо выбрать метки кластеров $\{y_i\}_{i=1}^N$ таким образом, чтобы расстояния между кластерами были максимальными. Центр $\boldsymbol{\mu}$ множества объектов $\mathbf{X}$ и центры кластеров $\{\boldsymbol{\mu}_k\}_{k=1}^K$ вычисляются по формулам:

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i \,; \quad \boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} [y_i = y_k] \mathbf{x}_i}{\sum_{i=1}^{N} [y_i = y_k]} \,. \tag{4.1}$$

Введем на множестве объектов $\mathbf{X}$ расстояние Махаланобиса

$$\rho(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{A}^{-1} (\mathbf{x}_i - \mathbf{x}_j)} \,, \tag{4.2}$$

где $\mathbf{A}$ — это матрица ковариаций множества $\mathbf{X}$

$$\mathbf{A} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top . \tag{4.3}$$

Определение 1. Функционалом качества кластеризации Q назовем межкластерное расстояние:

$$Q(\{\boldsymbol{\mu}_k\}_{k=1}^K) = \sum_{k=1}^{K} N_k \rho^2(\boldsymbol{\mu}_k, \boldsymbol{\mu}) \,,$$

где $N_k = \sum_{i=1}^{N} [y_i = y_k]$ — число объектов в кластере k.

Поставим задачу кластеризации как задачу максимизации функционала

$$Q\big(\{\boldsymbol{\mu}_k\}_{k=1}^K\big) \to \max_{\boldsymbol{\mu}_k \in \mathbb{R}^T}. \tag{4.4}$$

Для улучшения качества решения этой задачи предлагается применить метод метрического обучения к ковариационной матрице $\mathbf{A}$. Найдем такую матрицу $\mathbf{A}$, для которой функционал качества принимает максимальное значение:

$$\mathbf{A}^* = \arg\max_{\mathbf{A} \in \mathbb{R}^{T \times T}} Q\big(\{\boldsymbol{\mu}_k^*\}_{k=1}^K\big), \tag{4.5}$$

где $\{\boldsymbol{\mu}_k^*\}_{k=1}^K$ — решение задачи кластеризации (4.4).

4.2. Алгоритм адаптивного метрического обучения

Для решения поставленных оптимизационных задач (4.4), (4.5) используется алгоритм адаптивного метрического обучения. Предлагается понизить размерность пространства объектов $\mathbf{X}$ с помощью линейного ортогонального преобразования $\mathbf{G} \in \mathbb{R}^{T \times L}$, $\mathbf{G}^\top \mathbf{G} = \mathbf{I}$, где новая размерность $L < T$

$$\mathbf{X} \ni \mathbf{x}_i \mapsto \hat{\mathbf{x}}_i = \mathbf{G}^\top \mathbf{x}_i \in \mathbb{R}^L, \quad i = 1, \ldots, N.$$

Центр $\hat{\boldsymbol{\mu}}$ множества объектов $\{\hat{\mathbf{x}}_i\}_{i=1}^N$ вычисляется по формуле (4.1). Расстояния между объектами вычисляются по формуле (4.2), где в качестве матрицы $\hat{\mathbf{A}}$ используется матрица ковариаций (4.3) множества объектов $\{\hat{\mathbf{x}}_i\}_{i=1}^N$

$$\hat{\mathbf{A}} = \frac{1}{N}\sum_{i=1}^N (\hat{\mathbf{x}}_i - \hat{\boldsymbol{\mu}})(\hat{\mathbf{x}}_i - \hat{\boldsymbol{\mu}})^\top = \frac{1}{N}\sum_{i=1}^N \mathbf{G}^\top (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top \mathbf{G} = \mathbf{G}^\top \mathbf{A} \mathbf{G}.$$

Определение 2. Индикаторной матрицей назовем матрицу $\mathbf{F} = \{\delta_{ik}\} \in \mathbb{R}^{N \times K}$, где

$$\delta_{ik} = \begin{cases} 1, & \text{если } a(\mathbf{x}_i) = y_k; \\ 0, & \text{если } a(\mathbf{x}_i) \neq y_k. \end{cases}$$

Определение 3. Взвешенной индикаторной матрицей назовем матрицу $\mathbf{L} = \mathbf{F}(\mathbf{F}^\top\mathbf{F})^{-1/2} = \{l_{ik}\} \in \mathbb{R}^{N\times K}$, элементы которой равны:

$$l_{ik} = \begin{cases} \dfrac{1}{\sqrt{N_k}}, & \text{если } a(\mathbf{x}_i) = y_k; \\ 0, & \text{если } a(\mathbf{x}_i) \neq y_k. \end{cases}$$

Теорема 2. С использованием данных обозначений задача кластеризации (4.4) и задача метрического обучения (4.5) сводятся к общей задаче максимизации функционала качества [**?**]

$$Q = \frac{1}{N}\text{trace}(\mathbf{L}^\top\mathbf{X}^\top\mathbf{G}\hat{\mathbf{A}}^{-1}\mathbf{G}^\top\mathbf{X}\mathbf{L}) = \frac{1}{N}\text{trace}(\mathbf{L}^\top\mathbf{X}^\top\mathbf{G}(\mathbf{G}^\top\mathbf{A}\mathbf{G})^{-1}\mathbf{G}^\top\mathbf{X}\mathbf{L}) \to \max_{\mathbf{G},\mathbf{L}}. \tag{4.6}$$

4.3. Решение задачи метрического обучения

Для решения задачи (4.6) алгоритм адаптивного метрического обучения использует EM-подход. На каждом шаге итеративно вычисляются локальные оптимальные значения матриц $\mathbf{G}$ и $\mathbf{L}$. На E-шаге необходимо найти матрицу $\mathbf{L}$, которая является решением оптимизационной задачи (4.6) при фиксированной матрице $\mathbf{G}$. В качестве начального приближения получим взвешенную индикаторную матрицу $\mathbf{L}$ с помощью алгоритма кластеризации k-средних с евклидовой метрикой. На M-шаге производится нахождение оптимального значения матрицы $\mathbf{G}$ при фиксированной матрице $\mathbf{L}$. Алгоритм завершается при стабилизации функционала Q на последовательности итераций.

4.3.1. Алгоритм k-средних

В данной работе базовым алгоритмом для сравнения является алгоритм k-средних. Первым шагом алгоритм выбирает из множества $\mathbf{X}$ случайным образом K объектов $\{\boldsymbol{\mu}_k\}_{k=1}^K$ — начальные центры кластеров. Для каждого объекта $\mathbf{x}_i$ вычисляется расстояние (4.2) до каждого центра кластера $\boldsymbol{\mu}_k$ с единичной

матрицей трансформаций. Объект $\mathbf{x}_i$ относится к кластеру, расстояние до которого оказалось наименьшим. Далее производится вычисление новых центров кластеров по формуле (4.1). Алгоритм завершается, если значения центров кластеров прекращают меняться.

4.3.2. Оптимизация матрицы G с фиксированной матрицей L

Для любых двух квадратных матриц $\mathbf{A}$ и $\mathbf{B}$ справедливо $\text{trace}(\mathbf{AB}) = \text{trace}(\mathbf{BA})$. Данное свойство позволяет переформулировать задачу (4.6) следующим образом:

$$Q = \frac{1}{N}\text{trace}(\mathbf{L}^\top\mathbf{X}^\top\mathbf{G}(\mathbf{G}^\top\mathbf{AG})^{-1}\mathbf{G}^\top\mathbf{XL}) = \frac{1}{N}\text{trace}\big((\mathbf{G}^\top\mathbf{AG})^{-1}\mathbf{G}^\top\mathbf{XLL}^\top\mathbf{X}^\top\mathbf{G}\big).$$

Теорема 3. Обозначим $\mathbf{B} = \mathbf{XLL}^\top\mathbf{X}^\top$. Обозначим через $\mathbf{G} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]$ матрицу, состоящую из K собственных векторов матрицы $\mathbf{A}^{-1}\mathbf{B}$, отвечающих наибольшим собственным значениям. Тогда решением (4.6) является ортогональная матрица, полученная QR-разложением матрицы $\mathbf{G}$.

Функционал качества Q зависит только от матрицы $\mathbf{G}$. Обозначим

$$s(\mathbf{G}) = \text{trace}\big((\mathbf{G}^\top\mathbf{AG})^{-1}\mathbf{G}^\top\mathbf{BG}\big).$$

На данном шаге задача (4.6) принимает вид:

$$\mathbf{G}^* = \arg\max_{\mathbf{G}\in\mathbb{R}^{T\times L}} s(\mathbf{G})\,; \tag{4.7}$$

$$\mathbf{G}^\top\mathbf{G} = \mathbf{I}\,. \tag{4.8}$$

Ранг произведения матриц не превосходит рангов сомножителей, поэтому ранг матрицы $\mathbf{B}$ не превосходит K. Решением (4.7) является матрица $\mathbf{G} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]$, состоящая из K собственных векторов матрицы $\mathbf{A}^{-1}\mathbf{B}$, отвечающих наибольшим собственным значениям. Таким образом, размерность нового пространства объектов будет равна количеству кластеров K.

В общем случае матрица $\mathbf{G}$ не является ортогональной. Заметим, что для любой невырожденной матрицы $\mathbf{G}$ верно $s(\mathbf{G}) = s(\mathbf{GM})$. Для учета условия

ортогональности (4.8) найдем QR-разложение матрицы $\mathbf{G}$. Тогда ортогональная матрица $\mathbf{Q}$ является оптимальным значением $\mathbf{G}^*$.

4.3.3. Оптимизация матрицы L с фиксированной матрицей G

Теорема 4. Обозначим $\hat{\mathbf{K}} = (1/N)\mathbf{X}^\top\mathbf{G}\hat{\mathbf{A}}^{-1}\mathbf{G}^\top\mathbf{X}$. Тогда задача (4.6) эквивалентна задаче кластеризации k-средних с заданным ядром $\hat{\mathbf{K}}$ [?].

При фиксированной матрице $\mathbf{G}$ задача (4.6) принимает вид:

$$\operatorname{trace}(\mathbf{L}^\top\hat{\mathbf{K}}\mathbf{L}) \to \max_{\mathbf{L}\in\mathbb{R}^{N\times K}}.$$

Матрица $\hat{\mathbf{K}}$ является симметричной и неотрицательно определенной, тем самым может быть выбрана в качестве ядра.

4.4. Постановка задачи

Пусть объект $\mathbf{x}_i \in \mathbb{R}^n$ — временной ряд, последовательность измерений некоторой исследуемой величины в различные моменты времени. Пусть $\mathbf{X}$ — множество всех временных рядов фиксированной длины n, $Y = \{1, \ldots, K\}$ — множество меток классов. Пусть задана выборка $\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{\ell}$ — множество объектов с известными метками классов $y_i \in Y$.

Требуется построить точную, простую, устойчивую модель классификации

$$a : \mathbf{X} \to Y.$$

Данную модель представим в виде суперпозиции

$$a(\mathbf{x}) = b \circ \mathbf{f} \circ G(\mathbf{x}, \{\mathbf{c}_e\}_{e=1}^K), \tag{4.9}$$

где G — процедура выравнивания временных рядов относительно центроидов классов $\{\mathbf{c}_e\}_{e=1}^K$, $\mathbf{f}$ — алгоритм метрического обучения, b — алгоритм многоклассовой классификации.

4.4.1. Выравнивание временных рядов.

Для повышения качества и устойчивости алгоритма классификации предлагается провести выравнивание временных рядов каждого класса относительно центроида.

Пусть $\mathbf{X}_e$ — множество объектов обучающей выборки $\mathfrak{D}$, принадлежащих одному классу $e \in \{1, \ldots, K\}$. Центроидом множества объектов $\mathbf{X}_e = \{\mathbf{x}_i | y_i = e\}_{i=1}^{\ell}$ по расстоянию ρ назовем вектор $\mathbf{c}_e \in \mathbb{R}^n$ такой, что

$$\mathbf{c}_e = \underset{\mathbf{c} \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{\mathbf{x}_i \in \mathbf{X}_e} \rho(\mathbf{x}_i, \mathbf{c}). \tag{4.10}$$

Для нахождения центроида предлагается в качестве расстояния между временными рядами использовать путь наименьшей стоимости [**?**], найденный методом динамической трансформации времени. Псевдокод решения оптимизационной задачи (4.10) приведен в алгоритме 3.

Algorithm 3 Нахождение центроида DBA($\mathbf{X}_e$, n_iter)

Вход: $\mathbf{X}_e$ — множество временных рядов, принадлежащих одному и тому же классу, n_iter — количество итераций алгоритма.

Выход: $\mathbf{c}$ — центроид множества $\mathbf{X}_e$.

1: задать начальное приближение приближение центроида $\mathbf{c}$;
2: **для** $i = 1, \ldots,$ n_iter
3: **для** $\mathbf{x} \in \mathbf{X}_e$
4: вычислить выравнивающий путь между $\mathbf{c}$ и $\mathbf{x}$
alignment($\mathbf{x}$) := DTWalignment($\mathbf{c}, \mathbf{x}$);
5: объединить поэлементно множества индексов для каждого отсчета времени
alignment := $\bigcup_{\mathbf{x} \in \mathbf{X}_e}$ alignment($\mathbf{x}$);
6: $\mathbf{c}$ = mean(alignment)

DTWalignment($\mathbf{c}$, $\mathbf{x}$)

Вход: $\mathbf{c}, \mathbf{x}$ — временные ряды.

Выход: alignment — выравнивающий путь. // *каждый индекс временного ряда* $\mathbf{x}$ *поставлен в однозначное соответствие индексу временного ряда* $\mathbf{c}$

1: построить $n \times n$-матрицу деформаций DTW
cost := DTW($\mathbf{c}, \mathbf{x}$);
2: вычислить выравнивающий путь по матрице деформаций
alignment := DTWpath(cost);

Общая процедура выравнивания имеет следующий вид:

1) построить множество центроидов классов $\{\mathbf{c}_e\}_{e=1}^K$;
2) по множеству центроидов найти пути наименьшей стоимости между каждым временным рядом $\mathbf{x}_i$ и центроидом его класса $\mathbf{c}_{y_i}$;
3) по каждому пути восстановить выравненный временной ряд;
4) привести множества выравненных временных рядов к нулевому среднему и нормировать на дисперсию.

Результатом выравнивания должно стать множество выравненных временных рядов.

4.4.2. Метрическое обучение.

Введем на множестве выравненных временных рядов расстояние Махаланобиса

$$d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^{\mathsf{T}}\mathbf{A}(\mathbf{x}_i - \mathbf{x}_j)},$$

где матрица трансформаций $\mathbf{A} \in \mathbb{R}^{n\times n}$ является симметричной и неотрицательно определенной ($\mathbf{A}^{\mathsf{T}} = \mathbf{A}$, $\mathbf{A} \succeq 0$). Представим матрицу $\mathbf{A}$ в виде разложения $\mathbf{A} = \mathbf{L}^{\mathsf{T}}\mathbf{L}$. Матрица $\mathbf{L} \in \mathbb{R}^{p\times n}$ — матрица линейного преобразования, где p задает размерность преобразованного пространства. Если параметр $p < n$, то происходит снижение размерности признакового пространства.

Расстояние $d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j)$ есть евклидово расстояние между $\mathbf{L}\mathbf{x}_i$ и $\mathbf{L}\mathbf{x}_j$:

$$d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^{\mathsf{T}}\mathbf{L}^{\mathsf{T}}\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)} = \sqrt{(\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j))^{\mathsf{T}}(\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j))} = \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|_2$$

В качестве алгоритма метрического обучения в данной работе был выбран алгоритм LMNN. Данный алгоритм сочетает в себе идеи метода k ближайших соседей. Первая идея заключается в минимизации расстояний между k ближайшими объектами, находящимися в одном классе. Запишем функционал качества в виде

$$Q_1(\mathbf{L}) = \sum_{j \rightsquigarrow i} \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2 \to \min_{\mathbf{L}},$$

где $j \rightsquigarrow i$ означает, что $\mathbf{x}_j$ является одним из k ближайших соседей для $\mathbf{x}_i$. Вторая идея состоит в максимизации расстояния между каждым объектом и его объектами-нарушителями. Объектом-нарушителем для $\mathbf{x}_i$ назовем объект $\mathbf{x}_l$ такой, что

$$\|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_l)\|^2 \leqslant \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2 + 1, \quad \text{где } j \rightsquigarrow i. \tag{4.11}$$

Таким образом, необходимо минимизировать следующий функционал:

$$Q_2(\mathbf{L}) = \sum_{j \rightsquigarrow i} \sum_l (1 - y_{il}) \big[1 + \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)\|^2 - \|\mathbf{L}(\mathbf{x}_i - \mathbf{x}_l)\|^2\big]_+ \to \min_{\mathbf{L}},$$

где $y_{il} = 1$, если $y_i = y_l$, и $y_{il} = 0$ в противном случае. Положительная срезка позволяет штрафовать только те объекты, которые удовлетворяют условию (4.11).

Задача метрического обучения состоит в нахождении линейного преобразования $\mathbf{f}(\mathbf{x}) = \mathbf{L}\mathbf{x}$, то есть нахождении матрицы $\mathbf{L}$ в виде решения оптимизационной задачи

$$Q(\mathbf{L}) = \mu Q_1(\mathbf{L}) + (1 - \mu) Q_2(\mathbf{L}) \to \min_{\mathbf{L}}, \tag{4.12}$$

где $\mu \in (0, 1)$ — весовой параметр, определяющий вклад каждого из функционалов. Задача (4.12) представляет собой задачу полуопределенного программирования [**?**] и может быть решена существующими оптимизационными пакетами.

4.4.3. Классификация временных рядов.

Пусть $\mathbf{x} \in \mathbf{X}$ — неразмеченный временной ряд. Выравниваем временной ряд $\mathbf{x}$ относительно всех центроидов классов

$$\hat{\mathbf{x}}_e = G(\mathbf{x}, \mathbf{c}_e), \quad \text{где} \quad e = \{1, \ldots, K\}.$$

Отнесем временной ряд к классу, для которого минимально расстояние до соответствующего центроида. В качестве расстояния используем обученную метрику Махаланобиса с фиксированной матрицей $\mathbf{A}$

$$\hat{y} = \underset{e \in \{1,\ldots,K\}}{\operatorname{argmin}} \; d_{\mathbf{A}}(\hat{\mathbf{x}}_e, \mathbf{c}_e).$$

После нахождения оптимальных центроидов классов и нахождения оптимальной матрицы трансформаций процедура классификации заключается в измерении расстояния между найденными центроидами и новыми неразмеченными объектами.

Для оценки качества работы алгоритма будем вычислять ошибку классификации как долю неправильно классифицированных объектов тестовой выборки $\mathfrak{U}$:

$$\text{error} = \frac{1}{|\mathfrak{U}|} \sum_{i=1}^{|\mathfrak{U}|} [a(\mathbf{x}_i) \neq y_i].$$

Глава 5

Анализ прикладных задач

вставить эксперименты

Заключение

Литература

1. Hyonho Chun and Sündüz Keleş. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25, 2010.
2. Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118:62–69, 2012.
3. A. M. Katrutsa and V. V. Strijov. Stress test procedure for feature selection algorithms. *Chemometrics and Intelligent Laboratory Systems*, 142:172–183, 2015.
4. Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2017.
5. Paul Geladi. Notes on the history and nature of partial least squares (PLS) modelling. *Journal of Chemometrics*, 2(January):231–246, 1988.
6. Alexandr Katrutsa and Vadim Strijov. Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria. *Expert Systems with Applications*, 76:1–11, 2017.
7. Anastasia Motrenko and Vadim Strijov. Multi-way feature selection for ecog-based brain-computer interface. *Expert Systems with Applications*, 2018.